

Изменение имени сетевого интерфейса

Для Debian (и соответственно Ubuntu) подходит такой рецепт:

в файле `70-persistent-net.rules`, находящемся по адресу `/etc/udev/rules.d/70-persistent-net.rules`, в спецификации нужной сетевой карты изменяем в конце строки параметр `NAME=`, вписывая туда нужное имя интерфейса для сетевой карты с нужным MAC-адресом.

Например вместо `NAME=eth2` - `NAME=eth1`.

IGMP PROXY

Код:

```
wget
http://downloads.sourceforge.net/project/igmpproxy/igmpproxy/0.1/igmpproxy-0
.1.tar.gz?use_mirror=sunet
tar xzf igmpproxy-0.1.tar.gz
cd igmpproxy-0.1
```

Затем скомпилируйте и установите программу

Код:

```
./configure && make && make install
```

Теперь надо создать и отредактировать конфигурационный файл для программы, пример ниже

Код:

```
vim /etc/igmpproxy.conf
```

Содержимое конфигурационного файла

Код:

```
##-----
## Enable Quickleave mode (Sends Leave instantly)
## Эту строчку раскомментируют только в том случае,
## если внутри всего лишь один клиент IP TV
##-----
#quickleave

##-----
## Configuration for eth0 (Upstream Interface)
## Здесь указаны параметры для публичного интерфейса,
```

```
## на который придет поток от провайдера
## и указаны сети, с которых возможно будут подключаться клиенты
##-----
phyint eth0 upstream ratelimit 0 threshold 1
altnet 10.0.0.0/8
altnet 172.20.20.0/24
altnet 192.168.0.0/24

##-----
## Configuration for eth1 (Downstream Interface)
## Типовая конфигурация для внутреннего интерфейса
## смотрящего в домашнюю сеть
##-----
phyint eth1 downstream ratelimit 0 threshold 1
```

После создания конфига запускаем демона командой

Код:

```
/usr/local/sbin/igmpproxy /etc/igmpproxy.conf
```

Добавляем в автозагрузку

```
igmpproxy /etc/igmpproxy.conf & в /etc/rc.local
```

Удаление старых ядер

Сначала нужно узнать, какое ядро используется в Ubuntu.

```
uname -r
```

Эта команда выведет версию текущего ядра Linux. Его нельзя удалять.

```
root@server:~$ uname -r
3.0.0-25-generic
```

Далее нужно вывести список всех пакетов с установленными ядрами.

```
dpkg --get-selections | grep linux-image
```

В результате мы увидим что-то подобное:

```
root@server:~$ dpkg --get-selections | grep linux-image
ii linux-image-3.0.0-14-generic 3.0.0-14.23 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-15-generic 3.0.0-15.26 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-16-generic 3.0.0-16.29 Linux kernel image for
```

```
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-17-generic 3.0.0-17.30 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-19-generic 3.0.0-19.33 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-20-generic 3.0.0-20.34 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-21-generic 3.0.0-21.35 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-22-generic 3.0.0-22.36 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-23-generic 3.0.0-23.39 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-24-generic 3.0.0-24.40 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-3.0.0-25-generic 3.0.0-25.41 Linux kernel image for
version 3.0.0 on x86/x86_64
ii linux-image-generic 3.0.0.25.29 Generic Linux kernel image
```

Для того, чтобы удалить ненужное ядро, нужно выполнить следующую команду (предварительно указав название нужного пакета из списка выше):

```
sudo apt-get purge linux-image-3.0.0-14-generic
```

После того, как мы удалим все ядра, которые хотим, не забудьте выполнить команду `update-grub2`, чтобы обновить конфигурацию загрузчика.

Шлюз

Предполагается, что операционная система уже установлена, на сервере который имеет 2 сетевых интерфейса и будет выполнять функцию шлюза.

eth0 — подключение к интернет. (может получать IP адрес динамически, может иметь статический)

eth1 — подключение к локальной сети, будет иметь статический IP 192.168.10.1 и маску 255.255.255.0

Настройка сетевых интерфейсов

Редактируем настройки сетевых интерфейсов:

```
vi /etc/network/interfaces
```

Настраиваем **eth0** (по которому осуществляется подключение к интернет)

Вариант №1- Получение IP по DHCP от провайдера:

```
auto eth0
iface eth0 inet dhcp
```

Вариант №2-Статический IP

```
auto eth0
iface eth0 inet static
address XXX.XXX.XXX.XXX
netmask 255.YYY.YYY.YYY
gateway XXX.XXX.XXX.XXX
dns-nameservers ZZZ.ZZZ.ZZZ.ZZZ
```

Где:

Вместо XXX.XXX.XXX.XXX вписываем IP адрес, который мы получили от провайдера

Вместо 255.YYY.YYY.YYY - вписываем маску подсети.

Ну и вместо ZZZ.ZZZ.ZZZ.ZZZ вписываем IP адрес DNS сервера.

Настраиваем **eth1** (по которому подключается локальная сеть)

```
auto eth1
iface eth1 inet static
address 192.168.10.1
netmask 255.255.255.0
```

interfaces, примерно, следующего содержания:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address XXX.XXX.XXX.XXX
netmask 255.YYY.YYY.YYY
gateway XXX.XXX.XXX.XXX

auto eth1
iface eth1 inet static
address 192.168.10.1
netmask 255.255.255.0
```

Перезапустим сеть:

```
/etc/init.d/networking restart
```

В локальной сети адрес шлюза и dns сервера, будет 192.168.10.1.

На тестовой клиентской машине, прописываем IP адрес в ручную.

Присваиваем клиенту:

```
IP 192.168.10.2
маска подсети 255.255.255.0
шлюз 192.168.10.1
DNS 192.168.10.1
```

пробуем с клиента пинговать 192.168.10.1 — запросы должны бегать нормально.

Переходим на наш шлюз.

DNSMASQ

Установим пакет **dnsmasq**, он необходим для перенаправления DNS запросов, вышестоящим серверам.

```
apt-get install dnsmasq
```

Маршрутизация пакетов

разрешаем перенаправление пакетов:

```
vi /etc/sysctl.conf
```

необходимо найти строку и снять с нее комментарий:

```
net.ipv4.ip_forward=1
```

Сохраняем изменения и выходим.

[Доп. инфо о перенаправлении пакетов:](#)

How to Enable IP Forwarding in Linux

Aug 1st, 2006 | Comments

By default any modern Linux distributions will have IP Forwarding disabled. This is normally a good idea, as most peoples will not need IP Forwarding, but if we are setting up a Linux router/gateway or maybe a VPN server (pptp or ipsec) or just a plain dial-in server then we will need to enable forwarding. This can be done in several ways that I will present bellow. Check if IP Forwarding is enabled

We have to query the sysctl kernel value net.ipv4.ip_forward to see if forwarding is enabled or not: Using sysctl:

```
sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

or just checking out the value in the /proc system:

```
cat /proc/sys/net/ipv4/ip_forward  
0
```

As we can see in both the above examples this was disabled (as show by the value 0). Enable IP Forwarding on the fly

As with any sysctl kernel parameters we can change the value of net.ipv4.ip_forward on the fly (without rebooting the system):

```
sysctl -w net.ipv4.ip_forward=1
```

or

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

the setting is changed instantly; the result will not be preserved after rebooting the system. Permanent setting using /etc/sysctl.conf

If we want to make this configuration permanent the best way to do it is using the file /etc/sysctl.conf where we can add a line containing net.ipv4.ip_forward = 1

```
/etc/sysctl.conf:  
net.ipv4.ip_forward = 1
```

if you already have an entry net.ipv4.ip_forward with the value 0 you can change that 1.

To enable the changes made in sysctl.conf you will need to run the command:

```
sysctl -p /etc/sysctl.conf
```

On RedHat based systems this is also enabled when restarting the network service:

```
service network restart
```

and on Debian/Ubuntu systems this can be also done restarting the procps service:

```
/etc/init.d/procps.sh restart
```

Using distribution specific init scripts

Although the methods presented above should work just fine and you would not need any other method of doing this, I just wanted to note that there are also other methods to enable IP Forwarding specific to some Linux distributions. For example Debian based distributions might use the setting:

```
/etc/network/options:  
ip_forward=no
```

set it to yes and restart the network service. Also RedHat distributions might set this using:

```
/etc/sysconfig/network:  
FORWARD_IPV4=true
```

and again restart the network service.

Regardless the method you have used once you have completed this you can check it out using the same method shown above:

```
sysctl net.ipv4.ip_forward  
net.ipv4.ip_forward = 1
```

```
cat /proc/sys/net/ipv4/ip_forward  
1
```

If the result is 1 then the Linux system will start forwarding IP packets even if they are not destined to any of its own network interfaces.

IPTABLES

Теперь, нам необходимо добавить правила для маршрутизации пакетов:

```
vi /etc/rc.local
```

Добавляем перед строкой **exit 0**

```
iptables -F  
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
iptables -A FORWARD -i eth1 -o eth0 -j REJECT  
iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Сохраняем изменения, и перезагружаем сервер.

```
reboot
```

From:
<https://docs.infomir.com.ua/> -

Permanent link:
https://docs.infomir.com.ua/doku.php?id=knowledge_base:ubuntu_server_configure

Last update: **2022/12/05 18:35**

